

---

# **Minnesota Courts Integration Services**

---

Technical Overview  
Version 2.0  
September 11, 2006

## Table of Contents

<b>1. PREFACE.....</b>	<b>3</b>
<b>2. REVISION HISTORY.....</b>	<b>3</b>
<b>3. CONTACTS.....</b>	<b>3</b>
<b>4. MESSAGING.....</b>	<b>4</b>
<b>5. COMMUNICATIONS .....</b>	<b>4</b>
<b>6. SECURITY.....</b>	<b>4</b>
<b>7. MESSAGE FORMAT .....</b>	<b>5</b>
<b>7.1. SOAP Envelope .....</b>	<b>5</b>
<b>7.2. SOAP Headers.....</b>	<b>6</b>
7.2.1. Headers of type Address .....	7
7.2.1.1. Websphere MQSeries Addresses .....	7
7.2.1.2. HTTPS Addresses .....	7
7.2.2. Headers of type Endpoint Reference .....	7
7.2.3. Headers of type Security/UsernameToken .....	8
<b>7.3. SOAP Body .....</b>	<b>9</b>
7.3.1. CourtXML Versioning.....	9
7.3.2. SOAP Faults.....	9

## 1. Preface

This document gives a technical overview of the Integration Services provided by the Minnesota Supreme Court. From now on these services will be referred to as “Integration Services” in this document. It provides general details on how the Integration Services work, and system requirements for accessing them.

The following standards are used as a part of the implementation of the Integration Services, and are referenced throughout this, and other Integration Services technical documents.

Standard	Prefix *	XML Namespace	Link
SOAP 1.2	“soap”	<a href="http://www.w3.org/2003/05/soap-envelope">http://www.w3.org/2003/05/soap-envelope</a>	<a href="http://www.w3.org/2000/xml/Group/">http://www.w3.org/2000/xml/Group/</a>
WS-Security	“wsse”	<a href="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd</a>	<a href="http://schemas.xmlsoap.org/specs/ws-security/ws-security.htm">http://schemas.xmlsoap.org/specs/ws-security/ws-security.htm</a>
WS-Addressing	“wsa”	<a href="http://schemas.xmlsoap.org/ws/2004/08/addressing">http://schemas.xmlsoap.org/ws/2004/08/addressing</a>	<a href="http://www.w3.org/TR/ws-addr-core/">http://www.w3.org/TR/ws-addr-core/</a>
WS-Eventing	“wse”	<a href="http://schemas.xmlsoap.org/ws/2004/08/eventing">http://schemas.xmlsoap.org/ws/2004/08/eventing</a>	<a href="http://www-128.ibm.com/developerworks/webservices/library/specification/ws-eventing/">http://www-128.ibm.com/developerworks/webservices/library/specification/ws-eventing/</a>
CourtXML		<a href="http://www.courts.state.mn.us/CourtXML/X.X.X">http://www.courts.state.mn.us/CourtXML/X.X.X</a> (X.X.X corresponds to the version level)	<a href="http://www.courts.state.mn.us/is/">http://www.courts.state.mn.us/is/</a>

\* The prefix reference is the one used in this document. Other prefixes can be used as long as the namespace is correct.

## 2. Revision History

Revision Date	Person	Description
10/28/2005	TB	Created.
04/10/2006	TB	Added soap actor.
9/11/2006	TB	Migrated to soap version 1.2

## 3. Contacts

Questions: If you have questions, please contact Tim Buchholz or Linda Emeott, co-team leads of the Integration Team, in the Information Technology Department, Minnesota Supreme Court.

- Tim Buchholz, 651-297-7599, [tim.buchholz@courts.state.mn.us](mailto:tim.buchholz@courts.state.mn.us)
- Linda Emeott, 651-282-2063, [linda.emeott@courts.state.mn.us](mailto:linda.emeott@courts.state.mn.us)

## 4. Messaging

The Integration Services are implemented as sets of asynchronous message exchanges. These exchanges fall into 3 categories:

- **Request/Reply:** A request message is sent to the service and a reply message is eventually returned to an address specified in the request message. This category is most often used for services that provide query access into one of the Supreme Courts systems.
- **Submit/Response:** A message is submitted to the service. This submission may result one or more response messages being returned to an address specified by the submitter in the submitted message. This category is used most often by services that are used to make updates to one of the Supreme Courts systems. The purpose for response messages is usually to inform the submitter that their submission was either successful or unsuccessful.
- **Publish/Subscribe:** A message (referred to as a notification message) is published by a service. Consumers that have previously subscribed to that message will have it sent to them to an address specified when their subscription was created. This category is most often used to publish business events to interested consumers.

## 5. Communications

Integration Services are supported over 2 different communications transports:

- HTTPS
- IBM Websphere MQSeries

All messages are required to be transmitted over secure communications channels. This is generally done using SSL (https or MQSeries Channels using SSL).

## 6. Security

Integration Services use secure communication channels to protect messages as they are being transmitted between servers.

Each message that is either sent to, or output from, an Integration Service contains credentials that identify the submitting system. At the time that your computer system is registered for access to Integration Services you will be provided with the credentials that you will need to send along with messages that are input to Integration Services (Requests and Submissions), and will be able to specify the credentials that you want returned to you in Reply, Response and Notification messages. Currently these credentials are in the form of a user name and password. See section 7.2.3 for a description of the format for these credentials.

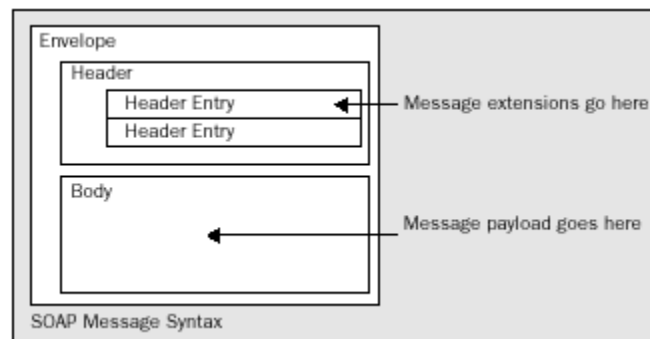
For input messages the Integration Services use these credentials to determine the rights to attribute to the sender. These rights include at a minimum the permission to access the particular service. They may also include others, such as permissions to certain court locations, lines of business or confidential access.

## 7. Message Format

### 7.1. SOAP Envelope

The overall format for Integration Services messages follows the SOAP (1.2) message structure. This is true for messages that are sent to an Integration Service and those that are output from an Integration Service.

SOAP messages are in XML format with an element named “Envelope” as its outermost element. The “Envelope” element contains a header section and a body section. The header section is used to hold message extensions (also referred to as message headers). Message extensions contain information that controls how messages are processed. The body section is used for application-specific data and is considered the payload of the message. This is where the information that is the actual purpose for sending the message resides.



**Figure 1: SOAP Message Structure**

```
<?xml version="1.0"?>
<soap:Envelope xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Header>
    .....
  </soap:Header>
  <soap:Body>
    .....
  </soap:Body>
</soap:Envelope>
```

**Figure 2: SOAP Message Syntax****7.2. SOAP Headers**

The header section contains XML elements known as headers. These headers contain control information about the message. The following is a list of the most common headers that may be included in Integration Service messages. See their associated specifications (wse, wsse, wsa, ...), and the sample messages, for details on their structure. See the technical documentation for each service for information on which headers apply to that service.

Header Element	Description	Type
wsa:MessageID	Contains a unique identifier for this given notification message.	String
wsa:RelatesTo	Contains the unique identifier (wsa:MessageID) from another message that this message is somehow related to.	String
wsa:Action	Contains a string that uniquely identifies the purpose for the given message. See the Integration Services technical documentation for each service for the specific values that will be/should be included in each messages.	String
wsa:To	Contains either an address that the message is being delivered or the name (Soap Actor) of the service that a message is being delivered to.	String
wsa:ReplyTo	Contains an address that should be used to return reply and response messages to. See section 7.2.2 for a description of headers of this type.	Endpoint Reference
wsa:FaultTo	Contains an address that should be used to return fault messages to. See section 7.2.2 for a description of headers of this type.	Endpoint Reference
wsse:Security/ wsse:UsernameToken	Contains information that will allow the receiver of a message to authenticate its sender. See section 7.2.3 for more information on this header.	UsernameToken
wse:Identifier	Contains an identifier that corresponds to the subscription that is being served by the delivery of notification messages. This identifier was assigned to the subscription when it was initially created.	String.
wse:Expires	<p>If the subscription that is being serviced by this notification message is set to expire, this header will contain the expiration date. There are 2 reasons that a subscription could be set to expire:</p> <ul style="list-style-type: none"> <li>It is for a version of CourtXML that is not current and an expiration date for that version has been set.</li> <li>An expiration date was specified when the subscription was created.</li> </ul> <p>When this date passes no more notification messages will be delivered based on this subscription. This is a mechanism to alert the consumer that their subscription is about to end.</p>	DateTime

### 7.2.1. Headers of type Address

Currently integration services can be accessed via 2 technical transports.

- Websphere MQSeries
- HTTPS

Addresses will be formatted differently based on the type of transport that is being used.

#### 7.2.1.1. Websphere MQSeries Addresses

When using Websphere MQSeries as the transport the addresses are formatted as follows:

wmq:QUEUENAME@QUEUEMANAGER

The first 4 characters are always “wmq:”. The rest of the address is made up of 2 components separated by an “@” character. The first component is a queue name. The second component is the queue manager that the queue resides on.

```
<wsa:To>wmq:MYQUEUE@MYQUEUEMANAGER</wsa:To>
```

**Figure 3: Sample wsa:To header for Webshpere MQSeries**

#### 7.2.1.2. HTTPS Addresses

When using HTTPS as the transport the addresses are formatted as follows:

https:webaddress

The first 6 characters are always “https:”. The rest of the address is made up of the URL.

```
<wsa:To>https:myaddress.com</wsa:To>
```

**Figure 4: Sample wsa:To header for HTTPS**

### 7.2.2. Headers of type Endpoint Reference

ReplyTo and FaultTo headers are of type Endpoint Reference and are used to specify the endpoint that should be used for reply and response messages. They will consist of a root element (named either wsa:ReplyTo or wsa:FaultTo) and the following child elements:

wsa:Address: Required, contains an address (see section 7.2.1).

wsa:ReferenceProperties: Optional, contains XML elements that the requestor/submitter wants returned as headers in reply/response messages.

```
<wsa:ReplyTo>
  <wsa:Address>wmq:MYQUEUE@MYQUEUEMANAGER</wsa:Address>
  <wsa:ReferenceProperties>
    <myafx:MyElement1 xmlns:myafx="mynamespace">MYDATA </myafx:MyElement1>
    <myafx:MyElement2 xmlns:myafx="mynamespace">THEDATA</myafx:MyElement2>
  </wsa:ReferenceProperties>
</wsa:ReplyTo>
```

**Figure 5: Sample wsa:ReplyTo header**

### 7.2.3. Headers of type Security/UsernameToken

Headers of type Security/UsernameToken are used by the IntegrationServices to identify and authenticate the system that is consuming the Integration Service. They contain a Username element and a Password element.

For messages that are sent as input to Integratiton Services the Username and Password values will have been assigned to you at the time you were given access to them. The Username is not specific to a specific user of your system, but is one value that corresponds to all users of your system when accessing Integration Services. You will use the same values no matter which service you are sending messages to. Integration services use these values to confirm that the sender has access to the service, and to determine what rights the sender has.

For messages output from Integration Services the values for Username and Password will be those that you specified when you requested access to integration services. The same values will be used no mater which Integration Service is sending you a message. You can use these values to confirm that the messages received come from a trusted source.

```
<wsse:Security>
  <wsse:UsernameToken>
    <wsse:Username">MyUsername </ wsse:Username>
    <wsse:Password">MyPassword</ wsse:Password>
  </wsse:UsernameToken >
</ wsse:Security>
```

**Figure 6: Sample wsse:Security/wsse:UsernameToken header**

When using HTTPS transport the components that you use to produce the UsernameToken elements will include many other elements and attributes that are not shown here.



At this time passwords are passed as plain text. This is one of the reasons that we require that all communications channels are set up using SSL encryption.

### **7.3. SOAP Body**

The soap body is where the actual payload of the message will be contained. The content is in XML Format and its elements are defined by the CourtXML Schema. See the Integration Services technical documentation for each service for a description of the message body for that service.

#### **7.3.1. CourtXML Versioning**

The version of the CourtXML schema that the body of a given message corresponds to will be shown by the namespace for the root element (first child of the soap:Body element). An example CourtXML namespace is:

<http://www.courts.state.mn.us/CourtXML/1.0.0>

This namespace indicates that the associated message is at version 1.0.0 of CourtXML.

To support version migrations Integration Services will support multiple versions at the same time. So, for instance, if version 2.0.0 of CourtXML is published, then existing messages will continue to be serviced at the 1.0.0 for a specified period of time. This will allow multiple partners to migrate to the new version of CourtXML as they are able, rather than be required to all migrate simultaneously.

Eventually old versions will expire and will no longer be supported. When this happens, any messages that are of that version of CourtXML will stop being processed.

#### **7.3.2. SOAP Faults**

If some kind of error has occurred as a result of a request or submission message the response and reply message bodies could consist of a soap fault. A common reason for receiving a soap fault is that the body of the corresponding request or submission message did not conform to an active version of the associated schema. Another is that the sender doesn't have the required access rights to use that service.

```

<soap:Fault>
  <soap:Code>
    <soap:Value>soap:Sender</soap:Value>
    <soap:Subcode>
      <soap:Value>soap:InvalidMessage</soap:Value>
    </soap:Subcode>
  </soap:Code>
  <soap:Reason>
    <soap:Text xml:lang="en">Message does not conform to schema.</soap:Text>
  </soap:Reason>
  <soap:Node>soap:Body</soap:Node>
  <soap:Role>http://www.w3.org/2003/05/soap-envelope/role/ultimateReceiver</soap:Role>
  <soap:Detail>
    <is:DetailString>The element 'CaseGetRequest' in namespace
'http://www.courts.state.mn.us/CourtXML/2.0.0' has invalid child element 'Selectionx' in namespace
'http://www.courts.state.mn.us/CourtXML/2.0.0'. List of possible elements expected: 'Selection' in
namespace 'http://www.courts.state.mn.us/CourtXML/2.0.0'.</is:DetailString>
  </soap:Detail>
</soap:Fault>

```

**Figure 7: Sample soap:Fault**

The following elements make up a soap 1.2 fault:

Element	Description
soap:Code/soap:Value	Code that shows where the error originated. Possible Values are: soap:Sender and soap:Receiver. The value soap:Sender indicates that the message had some problem with it prior to transmission and will have to be fixed prior to transmitting it again. It would not be appropriate to just retransmit the same message after receiving a soap:Sender fault. The value soap:Receiver indicates that a problem occurred after receiving the message and that the problem was not with the message itself. Retransmitting a message after a soap:Receiver message has been received may be successful.
soap:Code/soap:Subcode/soap:Value	Code that further describes the reason for the fault.
soap:Reason/soap:Text	Textual description for the fault.
soap:Node	The node within the message that the error resulted from.
soap:Role	URI indicating the role of the process that is returning the soap fault.
soap:Detail	Element containing some additional information that should help with resolving the fault.

The wsa:Action element for messages that contain soap faults will be the following:

<http://schemas.xmlsoap.org/ws/2004/08/addressing/fault>

The following table lists the soap faults that are common to integration services. See the document for the specific integration service that you are using for additional faults that may be specific to that service.

soap:Code/ soap:Value	soap:Code/soap:Subcode /soap:Value	soap:Reason/soap:Text	soap:Node	soap:Detail
soap:Sender	is:NotWellFormed	The Input Document is not well formed XML	soap:Envelope	String contain soap envelope.
soap:Sender	is:NotSoap12	The message is not version SOAP 1.2. Please reference the SOAP 1.2 namespace ( <a href="http://www.w3.org/2003/05/soap-envelope">http://www.w3.org/2003/05/soap-envelope</a> )	soap:Envelope	The input soap envelope.
soap:Sender	wsse:InvalidSecurityToken	An invalid security token was provided.	wsse:Security	The input soap envelope.
soap:Sender	soap:InvalidMessge	The message is invalid and cannot be processed.	soap:Envelope	The input soap envelope.
soap:Sender	soap:VersionMismatch	Cannot Determine Version Level.	soap:Body	The input soap envelope.
soap:Sender	soap:VersionMismatch	An Unsupported CourtXML Namespace was provided.	soap:Body	The input soap envelope.
soap:Sender	wsse:UnauthorizedAccess	Consumer does not have authorization to use service.	wsse:Security	The input soap envelope.
soap:Sender	soap:InvalidMessage	Message does not conform to schema.	soap:Body	The schema validation error information.
soap:Sender	wsse:MissingSecurityToken	No UsernameToken was provided.	soap:Header	The input soap envelope.
soap:Receiver	is:TechnicalError	<i>Some error message describing the technical error that has occurred.</i>	n/a	The input soap envelope.

